IN THE CLAIMS:

1. (Original) A method for compiling, in one of a plurality of compilation units, a subroutine having associated with it calls to items having addresses resolved external to said one compilation unit, said method comprising the steps of:

copying each of said external resolution items into said one compilation unit to form respective internal resolution items; and

compiling said subroutine using said external resolution items and said respective internal resolution items, each of said items having associated with it a respective version indicium for identifying inter-compilation unit version conflicts during execution of said compiled subroutine, wherein corresponding compiled external resolution items are executed in the case of inter-compilation version conflicts.

2. (Original) The method of claim 1, wherein said version indicium comprises at least one of a timestamp, a cyclic redundancy check (CRC) and a version control identifier.

3. (Original) The method of claim 1, wherein said internal resolution items are compiled to produce in-line executable code.

4. (Original) The method of claim 1, wherein at least one of said external resolution items comprise items resolved within a second of said plurality of compilation modules.

5. (Original) The method of claim 4, wherein said items resolved within said second of said plurality of compilation modules comprise copies of corresponding items resolved within a third of said plurality of compilation modules.

6. (Original) The method of claim 5, wherein said corresponding items resolved within said third of said plurality of compilation modules is executed in the case of inter-compilation version conflict.

7. (Original) The method of claim 1, wherein said external resolution items comprise Java® classes.

8. (Original) A method of retrieving a program for execution on a computer system, the program stored in a file in the computer system and including inter-compilation module calls, the method comprising:

comparing, for each inter-compilation module call having associated with it a cloned called entity and a respective external called entity, version identifiers of said cloned and external entities;

executing said cloned called entity in the case of said version identifiers comparing favorably.

9. (Original) The method of claim 8 wherein said version identifiers comprise at least one of a timestamp, a cyclical redundancy check (CRC) and a version control identifier.

10. (Original) A computer readable medium storing a software program that, when executed by a processor, performs a method for compiling, in one of a plurality of compilation units, a subroutine having associated with it calls to items having addresses resolved external to said one compilation unit, said method comprising the steps of:

copying each of said external resolution items into said one compilation unit to form respective internal resolution items; and

compiling said subroutine using said external resolution items and said respective internal resolution items, each of said items having associated with it a respective version indicium for identifying inter-compilation unit version conflicts during execution of said compiled subroutine, wherein corresponding compiled external resolution items are executed in the case of inter-compilation version conflicts.

11. (Original) The method of claim 10, wherein said version indicium comprises at least one of a timestamp, a cyclic redundancy check (CRC) and a version control identifier.

12. (Original) The method of claim 10, wherein said internal resolution items are compiled to produce in-line executable code.

13. (Original) The method of claim 10, wherein at least one of said external resolution items comprise items resolved within a second of said plurality of compilation modules.

14. (Original) The method of claim 13, wherein said items resolved within said second of said plurality of compilation modules comprise copies of corresponding items resolved within a third of said plurality of compilation modules.

15. (Original) The method of claim 14 wherein said corresponding items resolved within said third of said plurality of compilation modules is executed in the case of inter-compilation version conflict.

16. (Original) The method of claim 10, wherein said external resolution items comprise Java® classes.

17. (Original) A method for compiling, in one of a plurality of compilation units, a calling subroutine having associated with it called subroutines resolved external to said one compilation unit, said method comprising the steps of:

copying each of said external called subroutines into said one compilation unit to form respective internal called subroutines; and

compiling said calling subroutine using said external called subroutines and said respective internal called subroutines, each of said compiled external called subroutines having associated with it a respective version indicium for identifying inter-compilation unit version conflicts during execution of said compiled calling subroutine, wherein

corresponding compiled external called subroutines entries are executed in the case of inter-compilation unit version conflicts.

18. (Original) The method of claim 17, wherein said version indicium comprises at least one of a timestamp, a cyclic redundancy check (CRC) and a version control identifier.

19. (Original) The method of claim 17, wherein said internal called subroutines are compiled to produce in-line executable code.

20. (Original) The method of claim 17, wherein at least one of said external called subroutines is resolved a second of said plurality of compilation modules.

21. (Original) The method of claim 20, wherein said called subroutine within said second of said plurality of compilation modules comprises a copy of a corresponding called subroutine within a third of said plurality of compilation modules.

22. (Original) A framework for loading class data structures prior to execution and for resolving called Java® methods, said framework preferentially resolving said called Java® methods as cloned versions of Java® methods within a compilation unit common to a calling Java® method, said framework resolving respective called Java® methods outside said common compilation unit in the event of a version conflict between said respective cloned and external Java® methods.

23. (Original) The framework of claim 22, wherein said version conflict is determined with respect to at least one of a timestamp, a cyclic redundancy check (CRC) and a version control identifier.

24. (Original) The framework of claim 22, wherein said internal constant resolution entries are compiled to produce in-line executable code.

25. (Original) The framework of claim 22, an executing Java® method is provided addressability to a runtime version of its entry in a container class method table.

26. (Original) The framework of claim 23, wherein if a constant pool entry provided by said calling Java® method is to be resolved to a clone class, said framework performs the steps of:

loading said clone class; and

modifying said loaded clone class to represent the respective clone and parent classes for said constant pool entry.

27. (Original) The framework of claim 26, wherein said step of modifying comprises the steps of overlaying a plurality of fields within said clone class to represent corresponding structures of said parent class.

28. (Original) The framework of claim 26, wherein a determination of whether said constant pool entry provided by said calling Java® method is to be resolved to a clone class is made by performing the steps of:

extracting a corresponding constant pool entry pointer;

resolving the constant pool entry to its class; and

determining if the constant pool entry has been resolved to a clone class.